Automated Battleship

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Kyle John Kuennen

Fall, 2022

Technical Project Team Members

Maggie Bono

Deianira Griffith

Madisen Patrick

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Harry C. Powell Jr., Department of Electrical and Computer Engineering

The Sparks/Automated Battleship

Maggie Bono, Deianira Griffith, Kyle Kuennen, Madisen Patrick

December 13, 2022
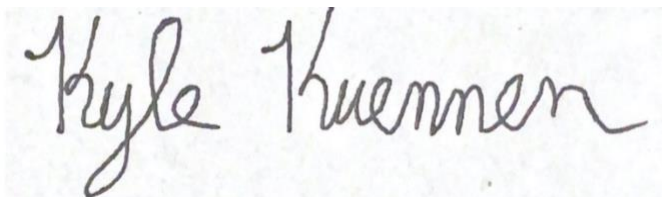
Capstone Design ECE 4440 / ECE4991

Signatures

Maggie Bono:

Deianira Griffith:

Kyle Kuennen:

Madisen Patrick:

Statement of work:

Maggie Bono

My primary responsibility was to design and create the physical device and parts of the game. I initially designed the device for the game in FreeCAD with plans to 3-dimensional (3D) print it but the 3D printer that was best suited for the design was broken. I decided to construct the device and a box to cover our power supply out of wood. I also created the grids that go on top of each array of Light-Emitting Diodes (LEDs) out of popsicle sticks, and I cut plexiglass to cover each Printed Circuit Board (PCB) so that the light from the LEDs could be seen. In FreeCAD, I designed cubes that the PCBs rested on and the ship pieces of the game. After 3D printing the cubes and ships, I glued magnets on the bottom of the ships. After each part was created, I assembled all the pieces together to create the final product.

My secondary responsibility was assisting Deianira with the design of the PCBs so I designed the power supply PCB in KiCad. I also created assembly diagrams and a bill of materials and sent those documents and the PCBs to WWW Electronics Inc. (3W) to have the LEDs, magnetic sensors, multiplexer chips, and multiplexing chips soldered on. After getting the boards back, I soldered the bypass capacitors on the PCB. In addition, it was my responsibility to choose the magnets to purchase and the initial testing of the magnetic sensors with the magnets.

Deianira Griffith

My primary responsibility was to design the schematic for the PCB, to create the initial PCB Layout, and the final PCB designs for the LED boards. I assisted Maggie in finalizing the Power Supply PCB and making any requested edits to the layouts of these boards. I spent the majority of the semester working on and refining these designs which were created from scratch. The schematic design was broken into various subsystems: the Power Supply, Hall Sensors Block, LED Matrices, Liquid Crystal Display (LCD) Display, and smaller subsystems.

Schematic designs for each of these subsystems was relatively simple and involved looking over datasheets to find components that fit our desired specifications. Once these components were found, the necessary and appropriate connections were made for these components within the schematic. Once the schematic was completed, I went to working on the PCB layouts for our LED Matrix boards. This process involved observing the manufacturing and budgeting limitations imposed on our project. My secondary responsibility was the hardware of the project. For this section I worked closely with Maggie to help her assembly the pieces she had made for the project into its final product. I also worked closely with Maggie on developing the PCB board for the Power Supply, and made the necessary adjustments to the design to allow for the needed connections between boards. As mentioned prior I focused heavily on finding the appropriate components for our project to be soldered in, and monitored their availability closely.

Kyle Kuennen

My primary responsibility was working on the embedded coding for the system. There were three main subsystems that had to communicate with our microcontroller: the Hall-Effect

sensor matrix, the two LED matrices, and the LCD display. All three subsystems were initially developed separately as well as tested to make sure they worked before being applied to the whole system. The hall-effect sensors were inputted into an 8:1 multiplexer and selection bits were cycled through code reading the inputs for all 64 sensors. The LED matrices were controlled through the use of Max7219 chips which interacted with the microcontroller through the use of the Serial Peripheral Interface (SPI) communication protocol. Similarly, our LCD was coded to communicate with the microcontroller through the Inter-Integrated Circuit (I2C) protocol. Once the three systems were working individually, I implemented them into the system as a whole making sure they all work accurately together.

My secondary responsibility involved assisting with the software regarding the game logic for the Battleship game. Madisen and I worked together to combine the peripherals that the microcontroller controlled with the logic and strategy system of the Battleship game. We then tested the entire code upon combination and made changes where necessary. Additionally, one of my other responsibilities included doing a power analysis of our entire system and ensuring we implemented an appropriate power supply and voltage regulators to meet our demands.

Madisen Patrick

My primary responsibility was coding the software concerning the Battleship game. I initially laid out the foundation of the code for the project by creating a high-level architecture. After creating the skeletal code for the embedded coding on Code composer Studio (CCS), I switched my focus to creating the game logic in code. Using C-Programming I created logic that manages the state of the game. This includes updating and controlling both the states of the player and computer's game boards and switching between Computer and player turns. The code also includes the computer non-deterministic Artificial Intelligence (AI), that plays the game against the player. When it is the computer's turn to move, the code makes an appropriate move based off of past moves. This code also controls when the game is over and who won the game.

My secondary responsibility was assisting with the embedded coding. I worked closely with Kyle to make sure that the embedded coding was communicating correctly with the game code. Also worked to make sure that the embedded coding information for reading the sensors corresponded to the appropriate game cells. LED output was tested to display the correct color and states of the games. We also worked together in debugging all of the code after combining the embedded coding with the battleship game software.

Table of Contents
Contents

Abstract

The proposed project is a game of Battleship with a fully automated opponent. The automated player will utilize an algorithm that will test a randomized location and will move orthogonally on its next move in response to a hit. Each coordinate on the game board will have magnetic sensors that will interact with corresponding magnetic sensors on the game pieces themselves. These sensors will allow the algorithm to automatically know if the location queried is a hit or a miss. Game progress will be tracked using Light-Emitting Diodes (LED) boards driven by multiplex chips. The LED will light up one color to indicate a hit, and another for a miss. One board will track the progress the automated player has made on the player, while the other will track the player's progress. The player will interface with the algorithm through an additional game piece with the magnetic sensor, the player will place this piece on the board to indicate their guess to the algorithm.

Background

The time spent playing video games since the pandemic has increased for 71% of players and most players reported an increase in playing in multiplayer mode [1]. The need for socialization increased due to the constriction of human interaction from the global lockdown after the start of the coronavirus pandemic. This technology allows the player to interact with the game in a similar manner as they would play this game with another human player. The thought behind this project is that it would be an enjoyable experience to build and a fun game to play after it is complete. Despite the pandemic being over for the most part, this project will allow individuals to feel less lonely by being able to play a game that allows you to feel like you're playing another individual. This project may not solve a notable problem, but it enables individuals to play a fun game against an automated player while being alone or isolated.

There are some publishing's online about individuals that have created an electronic chessboard using hall effect sensors [2]. The chess pieces have a small magnet in their base that triggers the hall effect sensors, and the microcontroller reads these sensors. This is a completely different game, but our game will also use hall effect sensors to read where the player has placed their ships. Hasbro's [3] has created an electronic battleship game. To input where you want your ships for this game, the player would press the buttons corresponding to the letter and number of where their ship is (B7, A3, etc.). To choose where to fire against the opponent, the player must push the buttons corresponding to the letter and number as well. Our game uses magnetic sensors to sense where the player has placed their ships which is different than Hasbro's buttons technique. This game is also a two-player game whereas our game is a one player game. Another difference is that our game uses LEDs to show whether the player has hit or miss versus Hasbro's game which uses sounds.

The knowledge gained from the previous courses in the engineering curriculum will guide us through the process of building the battleship game. Electrical and Computer Engineering (ECE) 3430 Introduction to Embedded Computer Systems, ECE 3501 and ECE

3502 Embedded Computing and Robotics I and II taught us embedded software which will help when implementing the Mixed Signal Processing 432 (MSP432) into our game. The embedded courses will also aid us in developing a test plan to ensure the software works. We gained knowledge on various electronic devices like magnetic sensors, circuit design, and PCB boards from ECE Fundamentals I, II, and III (ECE 2630, ECE 2660, and ECE 3750 respectively). Computer Science (CS) 3240 Software Development Methods expanded on software development techniques and how to structure the software in an optimal way.

Physical Constraints

Design Constraints
Our group, the Sparks, is made up of both Electrical and Computer Engineering majors. Thus, to fulfill the class requirements of the ECE 4440/4991 Capstone course, both a printed circuit board (PCB) and microcontroller need to be implemented into the design of the project.

CPU Limitations

There were to main limitations we needed to consider when selecting a CPU: memory and number of GPIO pins. Based off these considerations the MSP430F5529LP was selected. We anticipated using around 1-2kB of RAM and around 20kB of Flash memory. The MSP4305529LP allows for 8kB of RAM and 128kB of Flash. Secondly, the MSPF4305529LP has 40 GPIO pins readily available with our system being anticipated to only use 25 of those pins.

Software Availability

Code Composer Studio [4] was used to write the embedded coding and the software for the game because of its compatibility with the microcontroller. FreeCAD [5] was used to designing 3-D prints. KiCAD[6] was used to design the PCB boards.

Manufacturing Limitations

The PCB manufacturer, Advanced Circuits, imposed manufacturing constraints for the printed circuit board. These limitations were imposed to allow for a 2-layer board student special [7], they were required to have 62 mil thickness and the following specifications:

- Maximum board size: 60 square inches
- Minimum 5 mil line/space
- Minimum 10 mil hole size
- Maximum 50 drilled holes per square inch

Other constraints were that the housing for the PCB boards needed to be large enough to contain them, and the plexiglass grid combination.

The device of the game needed to be big enough to hold the LED matrix boards with a small amount of extra room. Also, the grids that were built to separate the light emitting from LEDs needed to be designed so that it fit the array of LEDs and the array of LEDs and sensors. Then the ships needed to be designed so that they could fit anywhere on the grid. The plexiglass

needed to be cut so that it fit inside the device and it was flush at the top of the device. The power supply box needed to be built so that the power supply PCB could fit into it as well as the tall wires connected to the PCB.

Parts Availability

With the Covid-19 Pandemic still affecting a majority of supply chains [8] the availability of our parts will be highly dependent on the stock of online suppliers such as Digikey [9] and Mouser [10]. Parts affected by this would be our multiplex chips, and magnetic sensors needed for board operation. We ran into a few supply issues throughout the project. While we foresaw this to be the largest and most prevalent constraint on our device, it ended up being one of the more minor ones for our group. We still had issues with this, in our voltage regulators, multiplexing and multiplexer chips. For our regulators it was difficult to find voltage regulators that met our specifications for both current and voltage (1 and 2 A and 3.3V and 5V respectively) at prices we could justify. Our Multiplexing MAX7219 chips were in a worryingly short supply towards our time of ordering, going from the 1,000 in stock from our initial consideration to 219 by the time we officially ordered them. Our Multiplexer chips were also difficult to find within our desired specifications, at the time of ordering there were not many digital 8:1 multiplexers in stock to order.

Tools Employed

Many tools were used to design, develop, assemble, and test our project. The tools for each category of our system are explained below.

Hardware

For circuit schematic and PCB board design the free software suite for electronic design automation (EDA), Kicad was employed. The suite has a connected schematic and PCB editor which allowed for quick and easy communication between the two. This allowed for us to quickly update the PCB should an edit have been made to the schematic. The program was new to us as we had used other EDA programs in our past, but we quickly took to the new software.

Once the PCB design was finished the online service FreeDFM [11] provided by Advanced Circuits was used to check it for any design errors that would prevent manufacturing. When this service returned with the "No Showstoppers" (nothing that would prevent manufacturing) screen we were allowed to put our board into production. The local circuit board assembly company 3W Electronics [12] were used to solder in our LED Arrays, Magnetic Sensors, Max 7219 Chips, and AH3582-SA-7 Multiplexer Chips.

Housing

The physical device of the game was created out of wood as well as the box to cover the power supply. To cut the pieces of wood to create the device and power box, a hacksaw and a table saw were used and glue was used to assemble the pieces of wood. The grids that separate the light emitting from the LEDs were created by cutting and gluing popsicle sticks together. FreeCAD was used to design the ships and cubes for the game because it is a good free option to

design 3D computer-aided-designs (CAD). To print out the designs made in FreeCAD, the UltiMaker S3 [13] 3D printer was used.

Software

Software tool used was Code Composer Studio which is a free software that allows you to integrate communication between a microcontroller. This tool was used to handle the communication between the game code and the embedded code. Existing libraries were used including the driver library of MSP430[14].

Societal Impact Constraints

Environmental Impact

The main environmental constraint of our project would be the manufacturing of it, and proper disposal once it has reached the end of its life span. It is unlikely that the pieces and parts used in the project will be recyclable, or not easily so. This could lead to more waste being produced in landfills should this become a common product. Our device will be powered via an electrical outlet. While this will cut down on waste created by lithium-ion batteries, there is still the concern of greenhouse gases emitted by the production of this electricity. [15] The production of electricity is not clean and simply powering the device has the potential to generate solid waste and to disrupt ecosystems by the land it takes up, etc. Tradition manufacturing methods are more wasteful than 3D printing but not all the materials used for the printing process are widely recyclable and some waste does end up in landfills or the natural environment [16]. To clarify, the plastics used to 3D print are recyclable but that is not an option everywhere. As 3D printing becomes more popular, the more options for recycling of this materials are available. 3D printing does reduce waste but still produces it. To manufacture PCBs and electrical components, dangerous chemicals are used which can poison the environment when discarded. PCBs are non-biodegradable and can last a long time [17]. PCBs and other electrical components release toxic substances in the ground, water, and air.

Sustainability

The system as a whole will be powered via an outlet in the wall, avoiding the need to continuously replace lithium-ion batteries. This is significant as both the manufacturing and disposal of these batteries negatively affects the environment. Additionally, the plastic used in the manufacturing of the structure of the project through 3-Dimensional (3D) printing can be easily recycled and repurposed. Through the lifetime of the system, the vast quantity of LEDs could potentially lead to some to stop lighting up. However, the design allows for any one LED to be easily replaced if needed as opposed to a whole array or strips of LEDs. To replace an LED, soldering equipment would be needed to desolder the impaired LED and to solder in a new LED.

Health and Safety

The primary concern for this device would be the choking hazard presented by the game pieces themselves. This device would not be suited for children 3 or younger, as they could easily choke on the game pieces posing a concern for their health and safety. [18] Warning would be placed on the device's packaging to denote such. Adult supervision would be required for assembly. This is to ensure that children do not harm themselves by catching their fingers between the magnetic pieces, or when closing the wooden gameboard. Another concern would be the electrical components of the device and ensuring that they do not cause a harmful discharge at any point. As per consumer standards [19] any electrical toy must be designed such that it can withstand damage to prevent shocking the consumer. To help remedy this each electronical component will be encased with proper external coverings. The device will also contain the proper warnings and labels as per consumer standards such as: to not submerge in water, shock hazard for the outlet, etc.

Ethical, Social, and Economic Concerns

Ethical Concerns

The device is rather expensive compared to its non-automated counterpart, which could cause an economical issue should the device become popular enough. Those from impoverished areas would be unable to afford and play with the device. This becomes an ethical issue as a once easily accessible and affordable game could become gentrified in a way, therefore removing the accessibility of the game to others. Besides this concern, there shouldn't be many others as our algorithm does not collect data on the demographics of its opponent. Rather the decisions made by it are based purely on the location of the game pieces themselves, which should not pose any ethical concerns in regard to profiling.

Social Concerns

Gaming has been associated with sleep deprivation, circadian rhythm disorders, depression, aggression, and anxiety when someone appears to be addicted to gaming. Gaming in moderation can improve the negative effects and most likely get rid of all the negative effects. According to an online survey, time spent playing games increased for 71% of respondents [20]. Feelings of loneliness increased by 5% during the coronavirus pandemic [21]. This battleship game mitigates feelings of loneliness by providing individuals with the feeling that they are playing a game with someone else even though the other player is an automated opponent.

Economic Concerns

Because this device is intended to help those struggling with loneliness, the price of the device should not be too expensive to reproduce. Should it be too expensive to reproduce the intended audience might not be able to afford it, by that people struggling with loneliness or parents looking to occupy their children.

External Considerations

External Standards

1. *Embedded C Coding Standard* – A book by Michael Barr explains the standards for software development process including comment rules and statement rules [22].
2. *National Electrical Manufactures Association (NEMA)* - Our project dealt with AC line and would fall under class 1 according to NEMA which refers to general purpose products. An enclosure around the electrical system was implemented to protect against dust, light, and indirect splashes of liquid [23].
3. *IPC* – Our product would fall under class 1 according to IPC, referring to everyday consumer electronics. IPC standards cover all aspects of PCB design including what are acceptable conditions, materials, layout, and assembly [24].
4. United States (U.S.) Consumer Product Safety Standard: Any product that is meant for children is subject to a set of federal safety rules. Specifically, Title 16 Chapter II Subchapter C Part 1505 details the requirements for electrically operated toys or games intended for use by children [19]. These, regulations relate to the project given that is an electronic game of Battleship that can be played by people of all ages, including children.

Intellectual Property Issues

This project could be patented based off there not being conflict with any patents that we could find. Additionally, it must be noted that the game of Battleship and the rules associated with it of which we took inspiration in our project are not able to be patented or copyrighted. The patents we looked at are described below.

The first patent to be discussed was issued in 1935 and presents a "Game Board" [25]. This patent describes one of the first iterations of the physical gameboard for the game we know today as Battleship. The main claim includes "A game board adapted to be positioned up right between two players and provided with two groups of perforations" and "plurality of pins adapted to be inserted through the perforations from opposite sides of the board in the playing of the game." There are some key differences between what is discussed in this patent versus our project. For instance, our game board does not incorporate any perforations or pegs but instead LEDs to display our hits and miss. Additionally, our board game design is only intended for one human player and is not intended for two opposing human players. Thus, even with the claims made by this patent, our project is still patentable.

The second patent to be discussed was issued in 2019 and presents a "Board Game with Dynamic Characteristic Tracking" [26]. The main claim includes "wherein each of the one or more game objects has a unique identifier and one or more characteristic values", "memory device that stores a unique identifier and associated with one of the game objects", and "a controller that via a wireless adapter adjusts the trait data of the characteristic values associated with one of the game objects." This patent mainly deals with dynamically tracking the

characteristics of an object. In our project our ships do have a unique identifier to differentiate between them each with their own traits. However, our project does not incorporate a wireless adapter to dynamically adjust the game objects themselves. Thus, even with the claims made by this patent, our project is still patentable.

Detailed Technical Description of Project

The goal of our project was to recreate the board game of battleship with a fully automated player. The user plays the game of battleship by communicating with the computer by placing the game pieces on the board. Each of the game pieces contain a magnet and the battleship board contains hall effect sensors in each gridded spot. Each time the computer or the player tries to make a guess, the board displays the move via LEDs. The player uses the LEDs and the LCD screen to see what move the computer player has made. The system design was broken down into the following sections:

Hardware
1. Power Supply
2. Player Communication Sensing System
3. Connection to MSP
4. Connection to LCD
5. LED Structure
6. Board Layout

Firmware
1. LED Control
2. Hall-Effect Input Detection
3. LCD

Software

1. Battleship Game
2. Computer Smart Logic
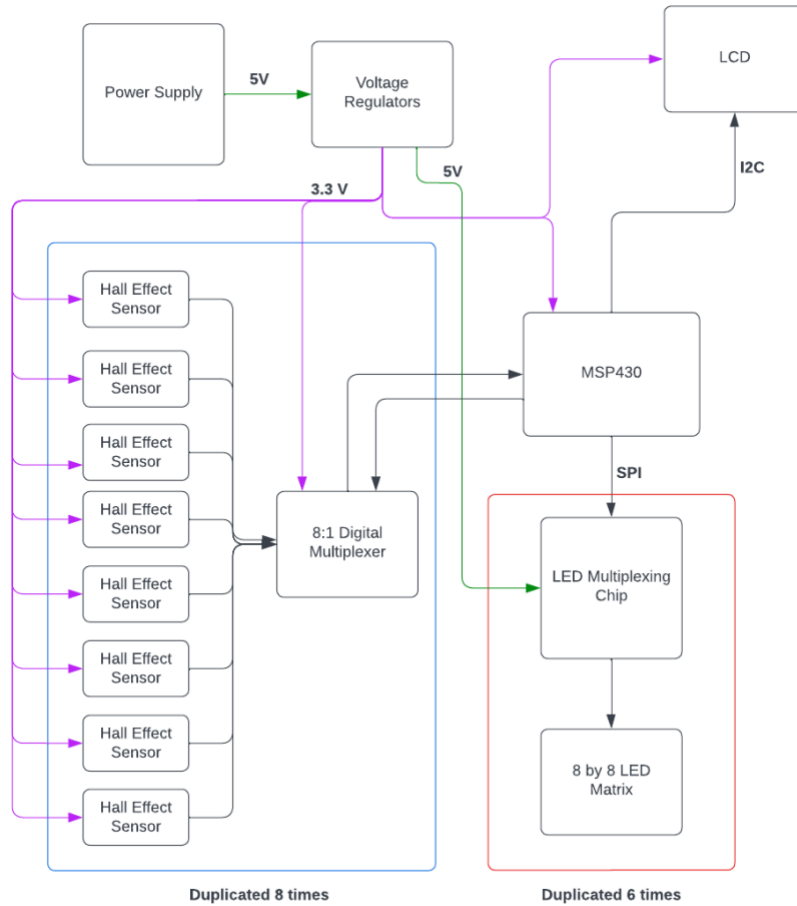
Mechanical

1. CAD Design
2. Assembly

Block Diagram

Figure 1 Block Diagram of Full System

Figure 1 displays the full system block diagram for the automated battleship device. The power supply plugs the device into a wall socket and regulates the voltage to supply the MSP430, Digital Multiplexers, and Hall effect Sensors with 3.3V, an additional voltage regulator to ensure a constant 5V was used to supply the multiplexing chips. The MSP controls the logic and interfacing with the other components of the system. The MSP first displays instructions on the LCD using SPI. The user then places their ships down on the grid and confirms their placement and the MSP reads and decodes the user's input. This input is then used to light up the appropriate LED through interfacing with the multiplexing chips.

Hardware

The hardware system is a PCB, that is composed of a power supply, connection to an LCD display, MSP432 microcontroller, 64 sensors, and 128 LEDs. The power supply powers the system with the LEDs and LCD displaying the game state and player interaction. The MSP430 controls the logic for the system and information from the sensors to control that logic. The hall

sensor blocks detect where the player has placed their ships, and where the player is guessing the computer's ships to be. A confirmation sensor can be found outside of the other blocks, this was used for confirming the player's selection in both ship placement and guess, allowing better control over both. The schematic was designed using KiCad and is shown in Figure 2. Each subsystem is described in detail in the following sections.
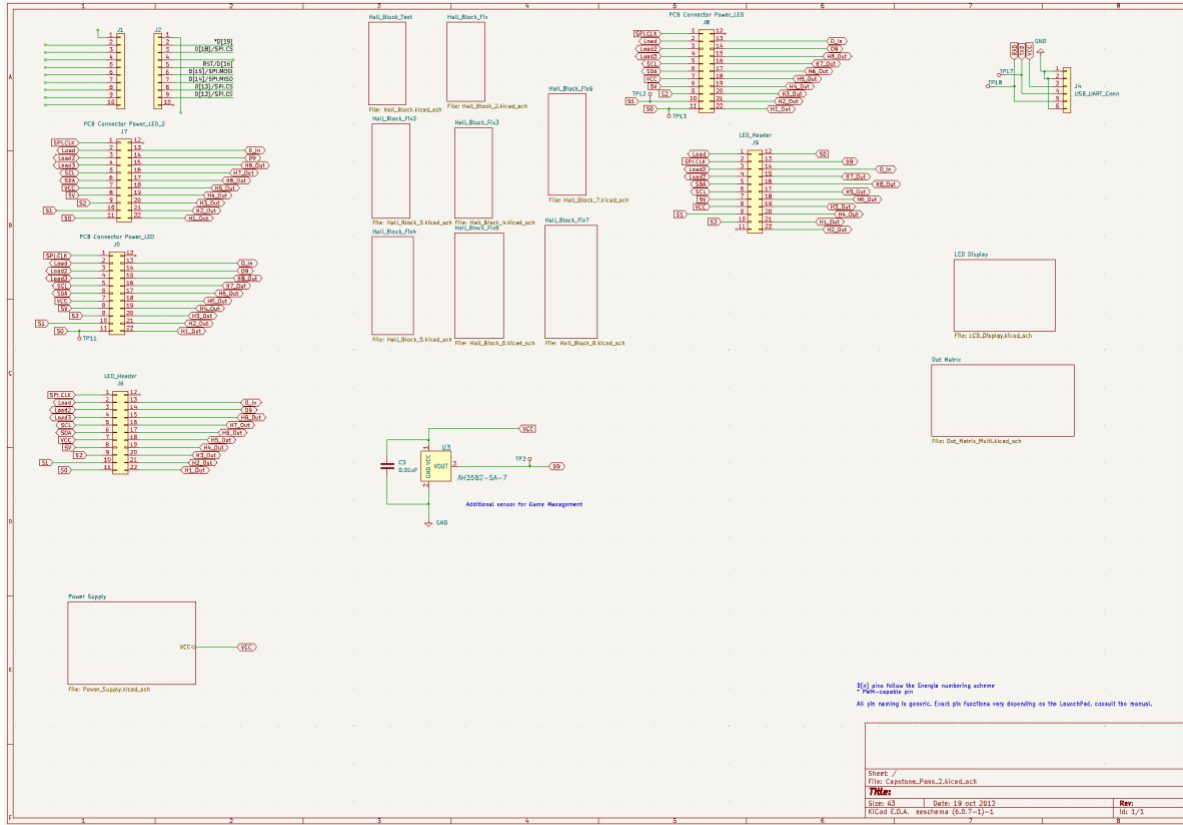

Figure 2 Overall Schematic of Battleship Board System

Power Supply

The Automated Battleship system is powered by a wall transformer that to a barrel jack connector on the PCB. Figures 3 and 4 display the power supply block and the hierarchical subsystem of the power supply.
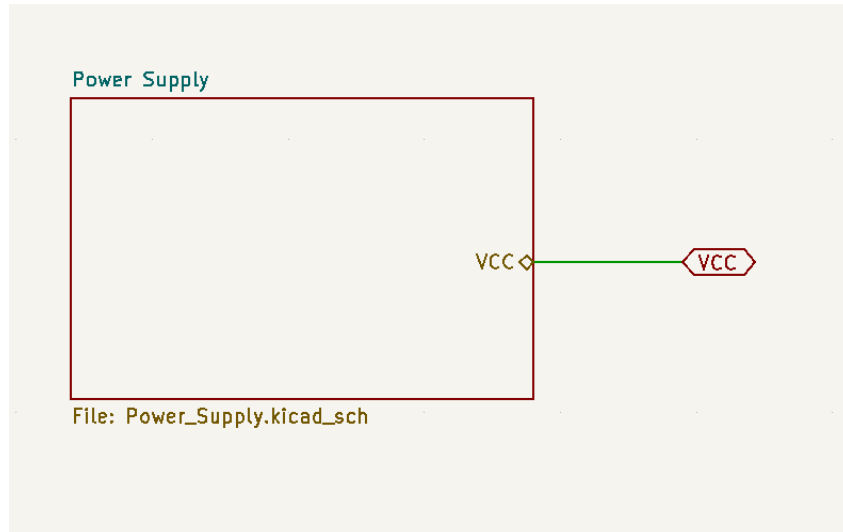
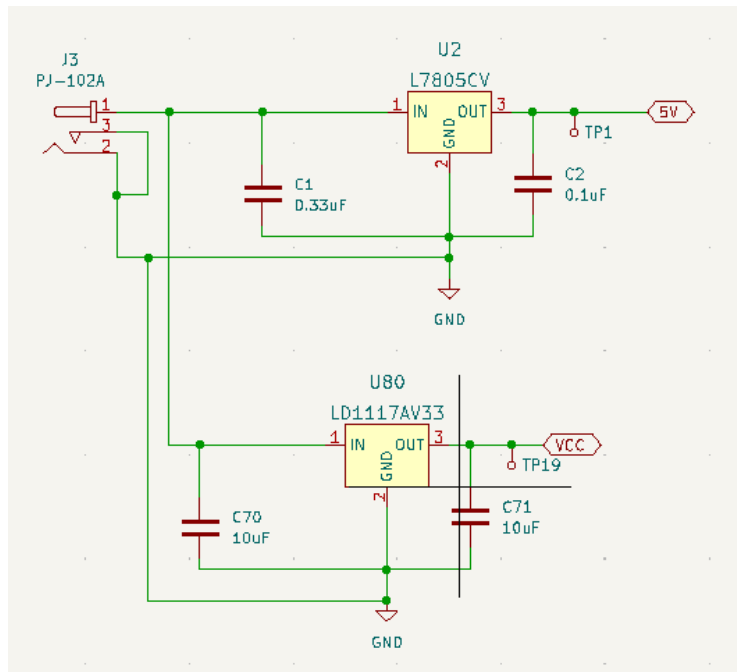**Figure 3 Power Supply Block**

Figure 4 Power Supply Schematic

Table 1 illustrates the current analysis of the system required for the voltage regulators L78S05CV [28] and LD1117AV33[29] where output is 5V and 3.3V.

3.3 Volt Supply	5 Volt Supply	Max Current (mA)	Max Power (W)
MSP430		100	0.33
Hall Effect Sensors (65)		455	1.5015
Digital MUX (8)		0.64	0.002112
	Multiplexing Chip (6)	1980	9.9
LCD Display		36.5	0.12045
Total		2572.14	11.854062 or 11.9

Table 1 Power Analysis of System

The total power required for the full system to run if every component drew full current is around 11.9W. Thus, a 15W AC/DC wall transformer that outputs 5V was selected to ensure enough power was available at all times. A 3.3V and a 5V voltage regulator were selected which could output 1A and 2A respectively to also meet our power requirements.

Player Communication Sensing System

Grid Locations

To allow for the player to communicate with the computer opponent, a system was needed to detect their inputs. We had decided early on to use hall effect (magnetic) sensors to achieve this due to their reliability over traditional buttons, which we were warned could have various issues with implementation. By placing hall effect sensors in each grid location, reading the combination of magnets or singular magnet placed, and decoding this within our code, we can determine where the player has placed their ship as well as where they are guessing the computer's ships to be.

Hall Effect

Hall Effect sensors were used to detect different magnet combinations. These sensors have three pins: VCC, Ground, and Output. The sensors will output a 0 or a 1 depending on the magnetic flux density (B) perpendicular to it. When the perpendicular magnetic flux is greater than the operation point ($B > B_{OP}$) the output is switched on (pulled low) and is held on until magnetic flux density B is lower than the release point ($B < B_{RP}$). The AH3582-SA-7 Digital Omni-polar Hall Effect Sensor [30] was selected because it supported a 3.3V input voltage, high temperature stability, enhanced immunity to stress, digital output, and the omni-polar nature allowed for less stress when designing the game pieces.

The greatest source of uncertainty revolved around the strength of the magnets. While the magnets had to be strong enough to be detected by the Hall Effect sensors through the plexiglass and added height of the grids, the magnets also had to be weak enough to not interfere with the

readings of other magnets within nearby or adjacent grid locations. If a magnet was too weak, it would not be detected at all by the sensor. On the other hand, if a magnet is too strong, nearby sensors corresponding to other grid locations would detect the signal and show that there was a magnet in place, even if there was not. Both of these scenarios would result in inaccurate readings, and would confuse our computer's algorithm.

To mitigate this issue an online magnet calculator was used. The magnet calculator calculated the amount of gauss produce by a specific magnet at a certain distance from the magnet. Using this calculator, multiple magnets were chosen to purchase. Multiple magnets were purchased to ensure that a magnet was found that is detected by the Hall Effect sensors but does not interfere with neighboring sensors.

Hall Effect Sensor System

Each hall effect sensor block represented a column of LEDs on the PCB. The schematic of these blocks can be seen in Figure 5.
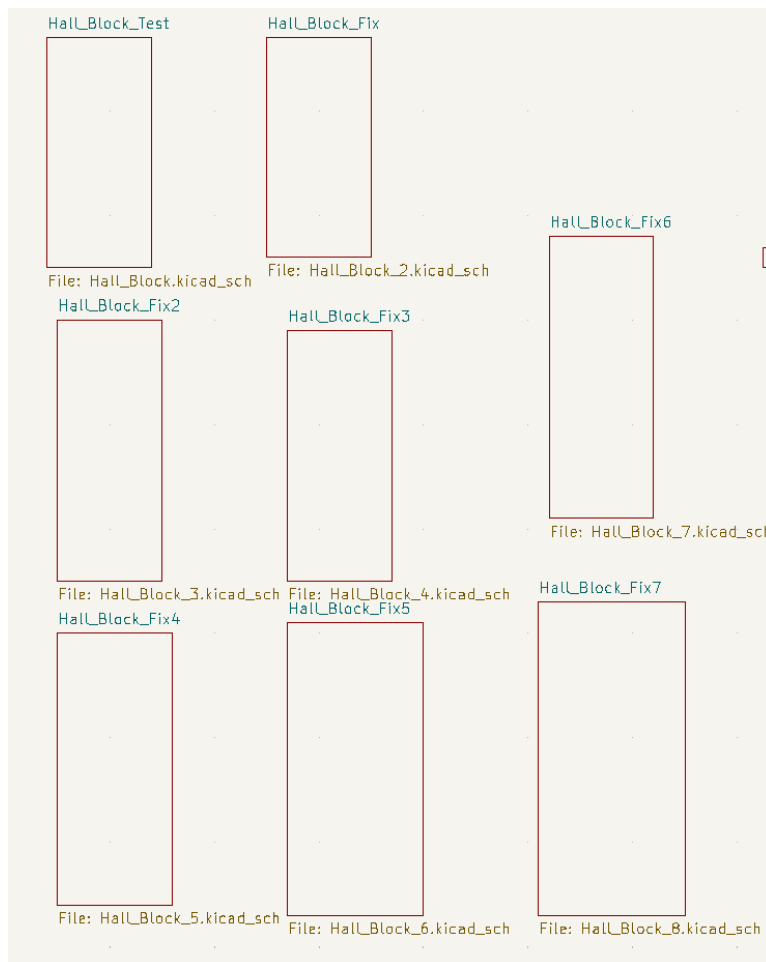

Figure 5 Hall Sensor Block

The hall effect sensor subsystem consisted of a multiplexer and 8 hall effect sensors, as shown in Figure 6. As mentioned prior, interaction with each block could consist of a singular magnet or multiple, a magnet could be placed on a singular grid location or across multiple when placing a ship. To allow for quick interaction with the game logic each hall effect sensor constantly outputted a 0 or a 1 to indicate its status. However, given the limited number of I/O pins on the MSP 430, the outputs of the hall effect sensors were connected to the 8 data inputs of a digital 8:1 Multiplexer chip [31]. Based on what values were sent to the A, B, C chip select lines, the output data of a different data input line to its singular output. Therefore, we used software to loop through each of the multiplexer inputs to read sensor's digital output. If the output data was a 0 it indicated the presence of a magnet, if it was a 1, we knew there was no magnet present. This allowed the game logic to proceed as expected with the correct inputs.

This subsystem was repeated 8 times to account for the 8 columns and rows of the sensors, totaling 64 sensors and 8 multiplexers. The schematic for each of these subsystems can be seen below in Figure 6.
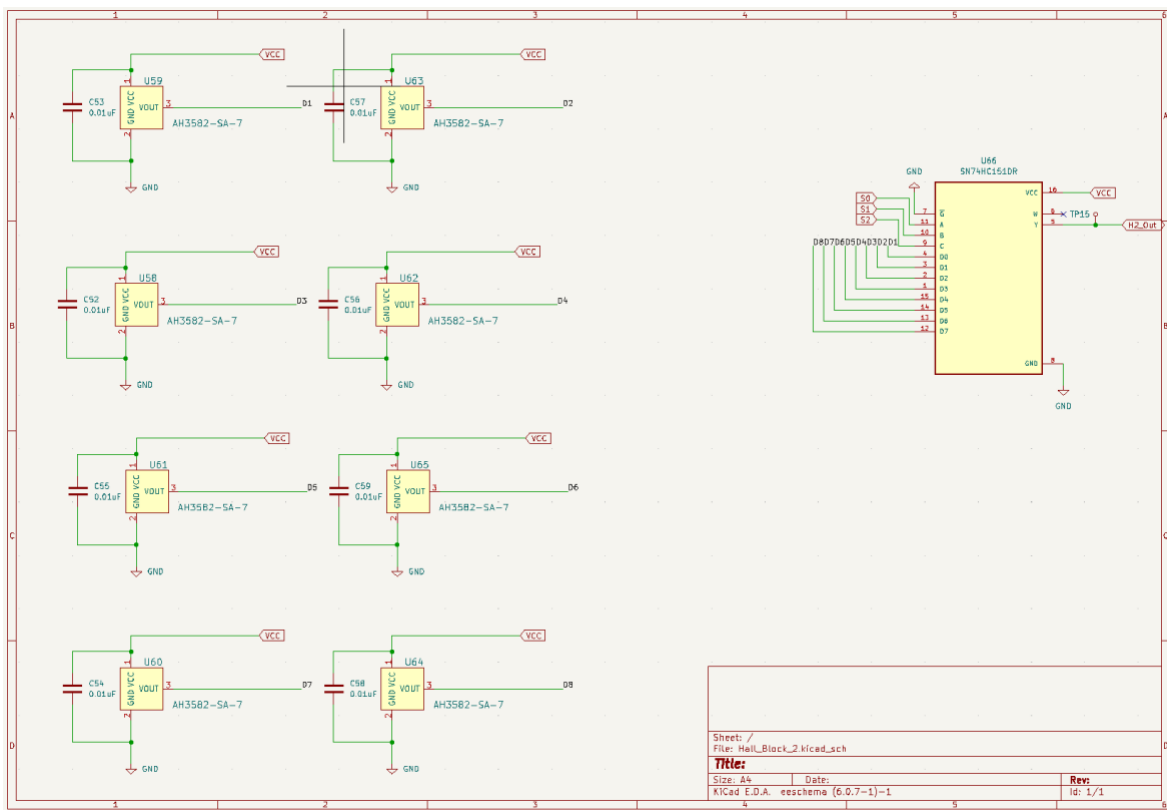

Figure 6 Schematic for One Row of Hall Effect Sensors

Connection to MSP430

The MSP 430 controlled all logic for the game. This microcontroller interfaced with the LCD, sensors and multiplexer blocks, and multiplexing chips that controlled our LEDs. Header pins were used to mount the MSP430 to our boards, and to allow for communication between our three boards and it. Wires were run from the MSP 430 to one of these header pins, which were connected to other sets of header pins. All needed connections for each subsystem were ran to these header pins to allow for each duplication amongst our boards.

Connection to LCD

Our chosen LCD, NHD-C0220BIZ [32] did not require an LCD connector and instead could be soldered directly into our PCB. With this in mind we made the necessary Serial Peripheral Interface (SPI) connections, and followed the data sheet for the recommended connections and component values. The schematic for our LCD can be seen in Figure 7.
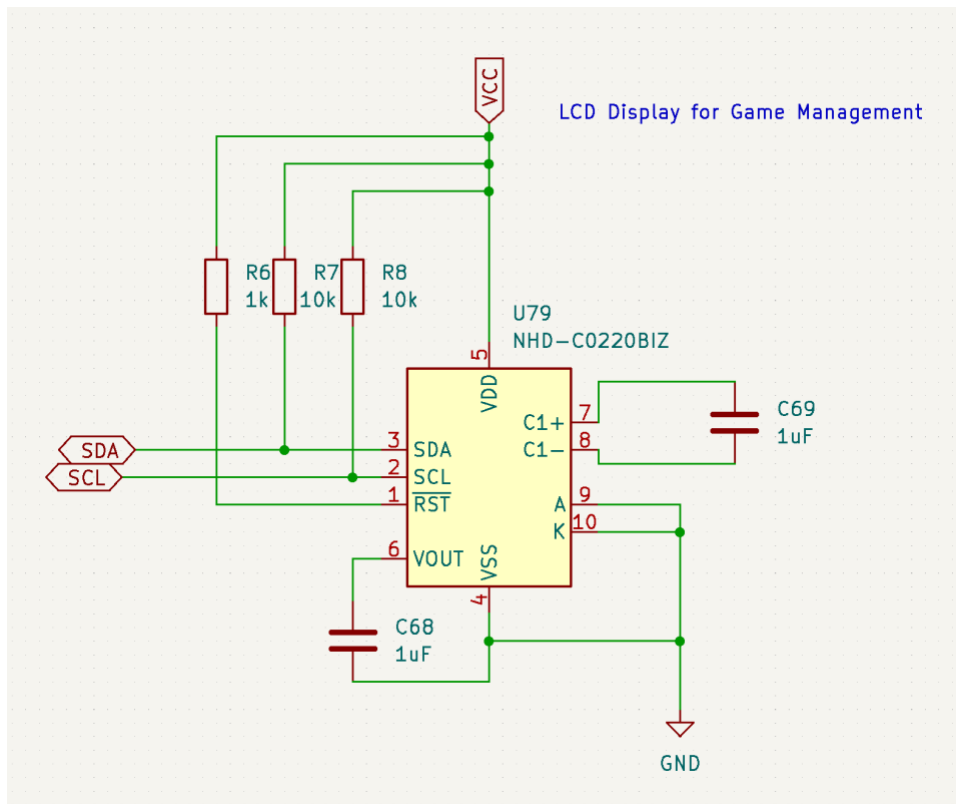

Figure 7 Schematic for LCD Display

LED Structure

Our LED subsystem consisted of 3 MAX7219 multiplexing chips [33] and an 8x8 array of CLX6F-FKC RGB LEDs with independent anodes and cathodes [34]. The MAX 7219 chips were selected due to their ability to effectively control a group of LEDs arranged in a grid pattern. The MAX7219 implements a LED multiplexing scheme which allows for independent control of an LED in a grid without having to have a separate GPIO pin from the microcontroller

19

for each one. The schematic for this subsystem can be seen in figure 8. For our LEDs to operate as intended with our game logic and MAX7219 chips, independent cathodes and anodes were needed. This decision greatly impacted our choice of LEDs. We eventually settled on the CLX6F-FKC RGB LEDs due to their independent anodes and cathodes, intense brightness, compact size, and price point. The latter of the two were considered greatly given the amount needed for our project (128) and the manufacturing constraints for our PCB boards. Individual LEDs were used instead of a premade matrix due to our desire to have the hall effect sensors close to the LEDs they would active and control. The use of independent LEDs allowed us to control the physical size of the matrix to account for the spacing needed for the hall effect sensors, and allowed us to place them within close proximity of the LEDs.
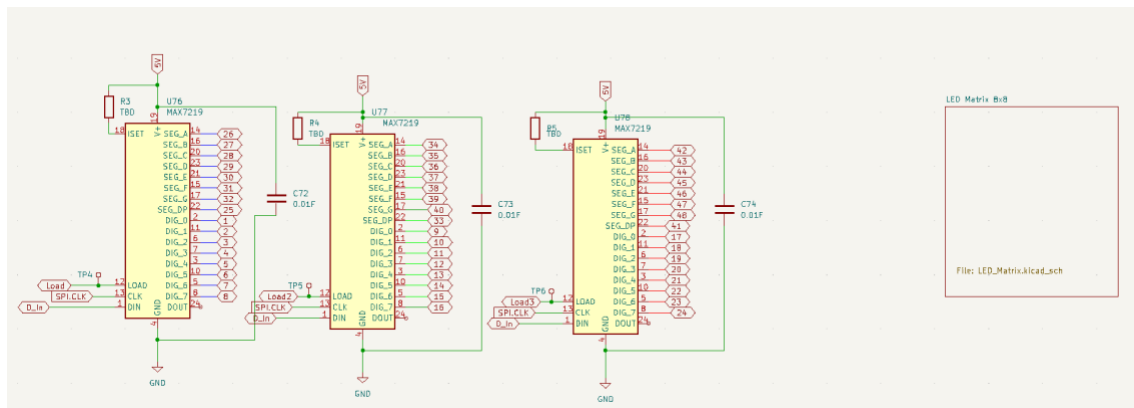

Figure 8 Schematic for Dot Matrix for LEDs

While independent anodes and cathodes were needed, within the LED matrix itself the anodes and cathodes of each color were connected horizontally and vertically. In other words, for one row of 8 LEDs the red anodes were connected to one another, as were the green and the blue. The same was done for each column with the cathodes—for one row of 8 LEDs the red cathodes were connected, as were the green and the blue. This configuration allowed the MAX 7219 chips to scan through the matrix, and allowed for each MAX 7219 chip to be assigned a specific color to control. The schematic for this LED Matrix can be seen in Figure 9.
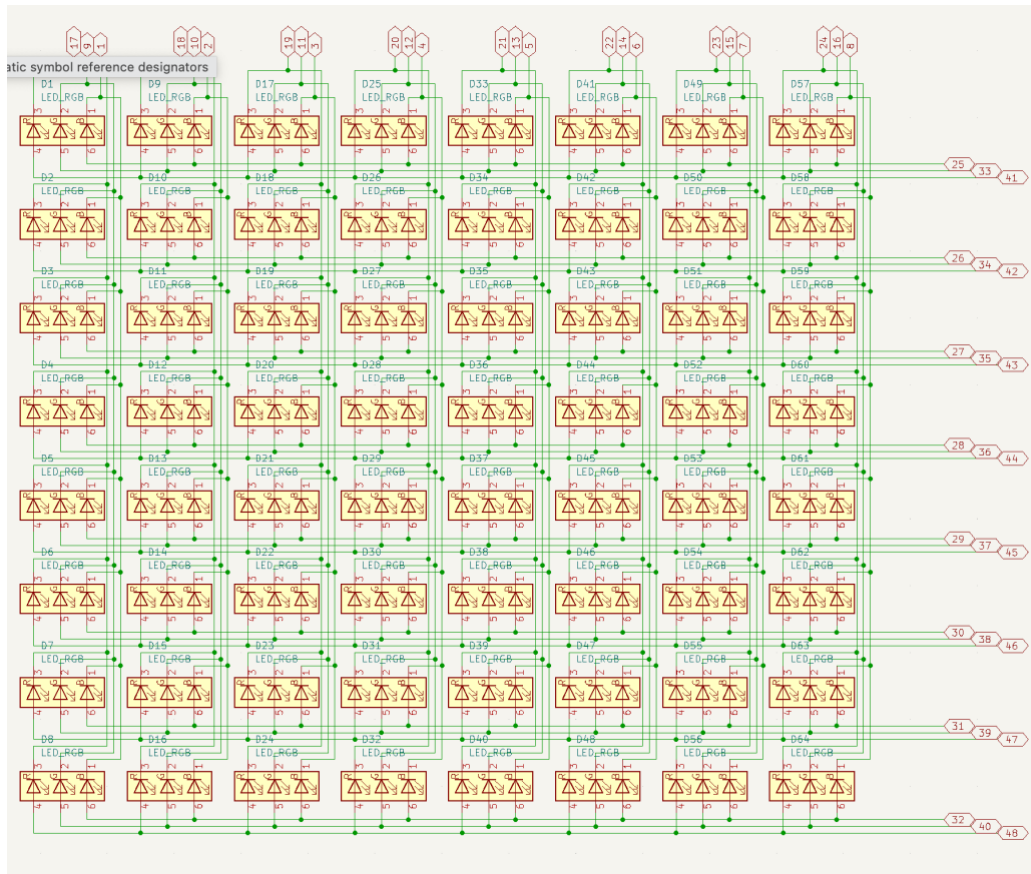
Figure 9 Hierarchical Schematic for LED Matrix

Board Layout

Three PCBs were designed to allow full functionality of this project. Each PCB including the reasoning behind specific design choices will be detailed below. Each PCB features holes in each of the four corners this was to allow for wooden dowels during assembly, as that was still in consideration at the time of design. Each board displays various test points throughout, all of which were placed closely with the Input/Output it was intended to test.

Power Supply

Three PCBs were designed for this project to allow for full functionality. The Power Supply PCB seen in Figure 10 was designed to allow the MSP430 to mount directly on it. There were no direct connections made between the mounting section and other sections of the board. This was done to account for any malfunctioning pins on our MSP430, allowing us to switch them easily. To accommodate for this lack of direct connection, 3 2x11 header pin configurations were placed on the board. The one closest to the MSP430 was used for direct wire connections between the MSP430 pins, and were labeled with their intended connections. The remaining header pin configurations were a mirrored version of MSP430 connection configuration. This was done to reduce the number of long traces needed to connect the 3 configurations. When

mirrored the configuration closest to the MSP430 connector only needed simple horizontal traces. These two mirrored header pin configurations were used to communication with the remaining two boards, and were connected to the boards via a ribbon cable. Finally for this board, the barrel jack was strategically placed towards the top of the board to allow for cables to be fed into the power supply's housing.
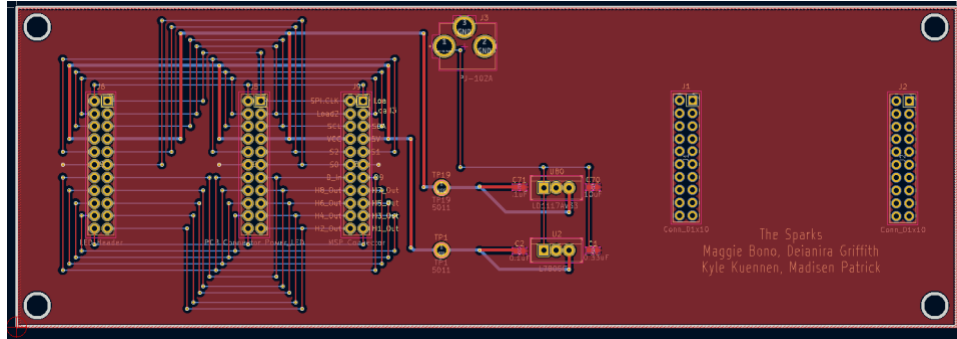

Figure 10 PCB Power Supply Layout

Player PCB

The Player's PCB as seen in Figure 11 was designed with the both the constraints imposed by the sensors and the board manufacturer. We needed our sensors to be far enough apart that they would not trigger one another due to the strength of the magnets needed, and close enough so that they would fit on the board. These constraints led to each sensor being placed horizontally, 0.4925 inches (12.510 mm) from one another on the first row, and vertically 0.5122 inches (13.010mm) apart from one another for the first and second row. All following rows followed the same spacing 0.4925 inches (12.510 mm) apart horizontally, and 0.6622 inches (16.820 mm) apart vertically. The LEDs were placed just under the sensors and were aligned horizontal to center with each sensor. The 8:1 multiplexer chip were placed at the bottom of each column, which reflects the design intended by the schematic.

The MAX 7219 chips were distributed to the side due to the space availability provided by the 2x11 header pin configurations. The right most of these configurations were placed at the bottom right of the board to allow for connection between the power supply, MSP430, and this board. The other configuration was to serve as additional test points, such as testing connections. The LCD display was placed beneath the LED matrix as to allow the player to quickly see game instructions while interfacing with the board. Finally, the confirmation sensor was placed close to the LCD as to not interfere with the other sensors, and to be near where the player would see instructions shortly after.
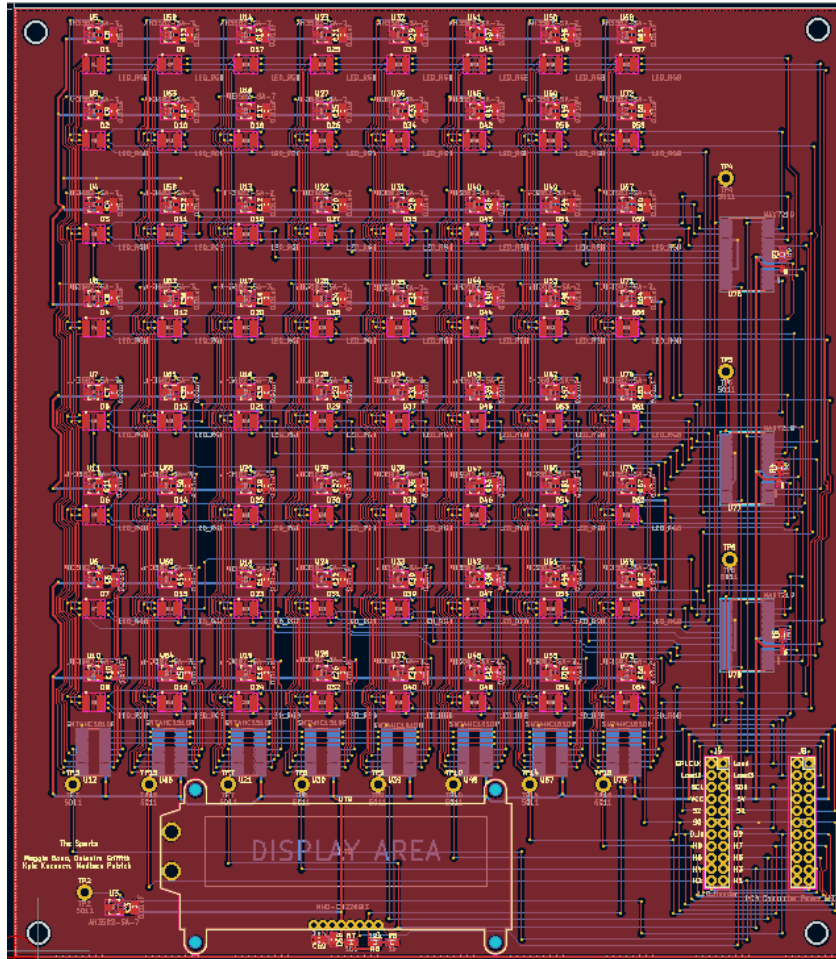
Figure 11 PCB Player LED Layout

Computer PCB

The PCB designed to display the computer's moves as seen in Figure 12, was largely the same as the one designed for the player. To keep a sense of consistency between the two board the PCB design for the player board was taken and all unnecessary components for the computer were stripped from it. This sped up the production of board designs as the LED placement and traces did not need to be tediously redone. One oversight in this method however was the need for each MAX 7219 chip to have its own Load line. By copying the design, the 3 load lines on this design were the same as the players. To accommodate for this when the components were soldered in, the header pins for Load, Load 2, and Load 3 on the 2x11 header pin configuration connecting to the power supply were not soldered in. The mirrored configuration designed solely for testing were soldered and wires were connected directly from the MSP430 to these pins. Thus, allowed for proper operation despite this oversight.
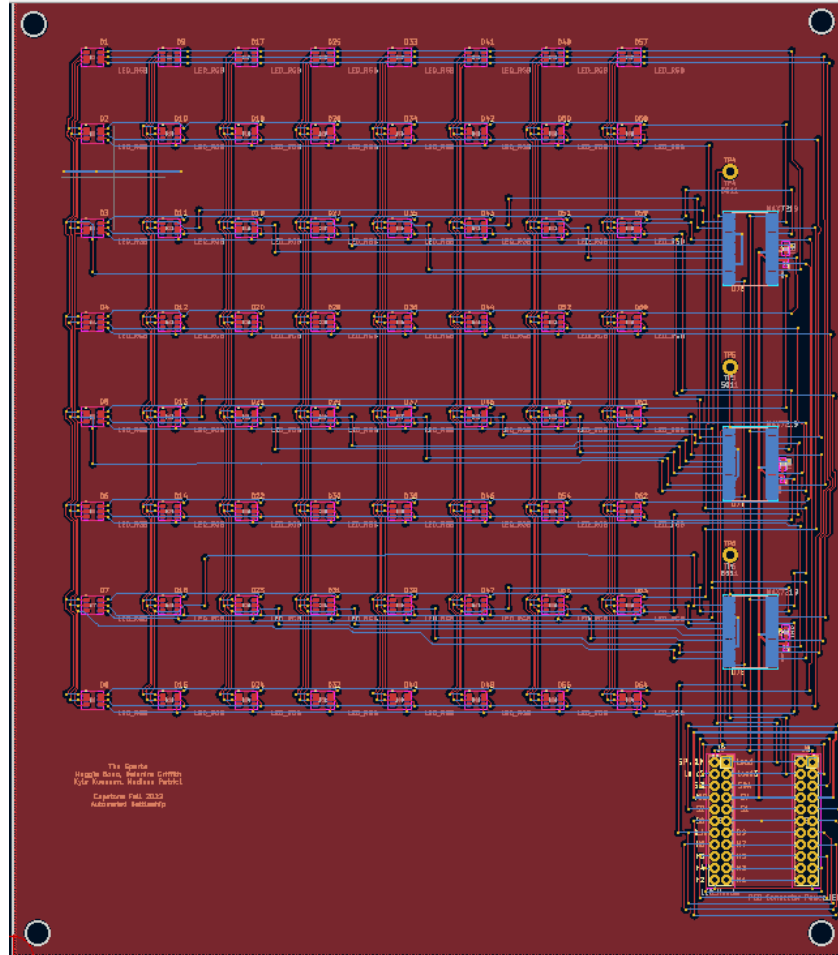
Figure 12 PCB Computer LED Layout

Firmware

Hall Effect Input Detection

The board is section off into an 8 x 8 grid that the player interfaces with. Each grid contains both an LED and a Hall Effect Sensor. In order to determine where a player has placed either it's ship or target selection while playing the game, each Hall Effect sensor had to be read. Each sensor has to be read to see if there is a magnet present. Both the ship pieces and the target selection piece contain magnets. If a magnet is found, the sensor outputs a low value with the output when there is no magnet being high.

There are 65 total magnetic sensors. One confirmation sensor which is read into the microcontroller directly through a pin on the microcontroller. The other 64 pins are interacted with through the use of eight 8:1 multiplexer. These multiplexers have three select pins corresponding to eight possible combinations in binary: 000, 001, 010, 011, etc. The code and the microcontroller loops through each of these combinations and then the output is taken in and

stored. This process occurs through the use of a timer and interrupt service routine allowing for the data from the sensors to be constantly read in throughout the length of the game.

LED Control

        The board contains 128 RBG LEDs: 64 LEDs represent the Player's Board and 64 LEDs represent the Computer's Board. In the beginning of the game both boards LEDs are off. Once the player has placed its ships and the game begins. All LEDs on the computer's board are made set to blue. All the LEDs on the Player's board are set to blue except the cells that have a ship in it, which are set to display green. Each time a player has selected a move, a singular LED is then updated with the appropriate color: red for a hit or white for a miss. Once the game is over, the board lights all LEDs to green or red. If the player wins the game the LEDs all flash Green. If the computer wins the game the LEDs all flash Red.

        The LEDs were interfaced through the use of six Max7219 multiplexing chip. The chips were controlled through the SPI communication protocol. There were six separate chip select lines corresponding to the six different Max7219 chips. To control a specific LED, the chip select line was first set to low. Then the address was sent which corresponding to the row of the LED matrix you want to change. Then data was sent regarding the columns that are needed to be toggled. This is followed by setting the chip select line high for the specific Max7219 chip that data should be loaded into. This chip select line is then set back low. Thus, the data is sent to all six chips, however the toggling of each chip's chip select line determines if that data is loaded into it effectively allowing for communication to each individual chip.

LCD

        The LCD screen communicates to the player the state of the game. It notifies the player: when to place its ships on the board, the computer's turn, the result of the computer's selection, the player's turn, and who won the game. Other actions that had to be coded with regard to the display included it being initialized and the display being cleared.

        The LCD is controlled by the microcontroller through a software implementation of the I2C communication protocol. In order to output to the display, it first receives the start sequence. This involves setting both the serial data line (SDA) and serial clock line high (SCL) and then setting them both low. Following this, the address of the LCD display is sent, in this case the address was 0x78. The control byte was then transmitted. There are multiple control byte combinations but the one that allows for the sending of words to the display is 0x40. The text is then sent followed by the stop command. The stop command is where the SDA and SCL lines are first set to low and then both are set to high.

Software

Battleship Game

The battleship game state and information are handled in the C code. It mainly holds the state of the Player and Computer game boards. Theses gameboards are 8 x 8 matrices that hold cell information, the coordinate and the state of the cell. The game is started by initializing both boards, by making the state of each cell water. The computer's ships are then selected randomly and its game board is updated accordingly by changing the state of the cells to ship occupied. The same process is used once the player has placed it ships and the information from the magnet game pieces has been translated.

Once both player's ships have been placed, the game begins by randomly selecting which player goes first. If the player is selected to go first the program waits for a target to be selected. Once a target is selected, it is then verified to make sure that the move is valid and checks the location on the computer's game board. If the state of the cell on the game board is water, the cell state is then changed to a miss. If the state of the cell is ship occupied, the cell state is then changed to a hit. After the selection it is then changed to the computer's turn. Once the computer has selected a cell the same process that was used for the player target selection is then repeated.

After one player has selected all the opponents ship locations, a winner is then determined. After the winner is notified, it restarts the game.

Computer Smart Logic

The computer smart logic plays against the human player. The computer's four ships were coded to randomly select valid positioning for the ships. A valid positioning for the four ships was determined by first randomly assigning the direction the ship would be placed either horizontal or vertical. Once the direction of that ship is assigned a position is then based on the direction and length specified for each ship. If the direction is horizontal, a random number for the row is assigned between 0 and 7 and a random number for the column is assigned between 0 and 7 minus the length of the ship. If the direction is vertical, a random number for the column is assigned between 0 and 7 and a random number for the row is adjusted to be between 0 and 7 – the length of the ship. Once a placement has been assigned, the position is then verified to ensure that all locations were on the board and not overlapping. After the ship placements are then saved into the computer's game board array.

Once the game has begun and it is the computer's turn, it chooses between two modes. The first mode, hunt mode, randomly selects cells as targets and checking if the location is water. When a cell equates to a hit, then hunt mode is set to false and the second mode is set to true. The second mode then selects all possible cells around the hit location. Once it finds another hit, the computer then continues to select cells in that column or row until it has sunken the ship. Once the ship has been sunk, hunt mode is then activated again.

Mechanical

Device

The game device was initially designed in FreeCAD to fit the LED matrix PCBs with the plan to 3D print the design. There was only one 3D printer that was free of charge that could be used to print the design which was the Lulzbot Taz 6. This printer had a heated bed which would be optimal for the design due to the high surface area of the device and this printer had a bed big enough to be able to print the design. The Lulzbot Taz 6 printer was broken and could not be used and the only other option that had a heated bed and a large enough bed was the Lulzbot Taz 6 Flexistruder printer. It was not optimal for the device to be printed with the flexistruder printer because we wanted the device to be sturdy. It was decided that the best option was to build the device out of wood. The requirement for the size of the box was that it had to be big enough to fit the LED matrix PCBs inside it and with extra room on each side of the PCBs and under the PCBs. The device was made by cutting pieces of wood with a table saw and the pieces were assembled with glue. After the device was put together, it was spray painted black. The device before it was painted is shown in figure 13.

Figure 13 Device of the Battleship Game

A box was also built out of wood, as shown in figure 14, and was designed to be able to fit the power supply PCB. The box was made by cutting pieces of wood with a hack saw because that was the only saw available at the time and glued together. The box was also designed so that it did not exceed the width of the device so that it could be attached to the back of the device by glue while being aligned with the width of the device, as shown in figure 15.

Figure 14 Power Supply Box that Holds the Power Supply PCB

Figure 15 Picture of Device and Power Supply Box Together

CAD Design

Cubes were designed in FreeCAD with the purpose of holding the LED matrix PCBs so that they were lifted from the surface of the device as shown in figure 16. They were also designed with a circular pocket to fit a wooden dowel. Initially the wooden dowel would hold the PCB to cubes. It was forgotten to drill the holes into the PCB before soldering started so it was decided to just glue the corner of the LED matrix PCBs to the cubes. The purpose of lifting the PCBs above the surface was to allow for room for the connections between the boards and to reduce the distance between the magnets and the magnetic sensors. The size of the cubes was designed specifically to set a specific distance between the magnets and magnetic sensors so that the magnets were sensed by the sensors without error while ensuring that there would be room for the connections. The cubes were 3D printed using the Ultimaker S3 printer because this

printer produces a more detailed print than the other options that were available to print them with no financial cost.
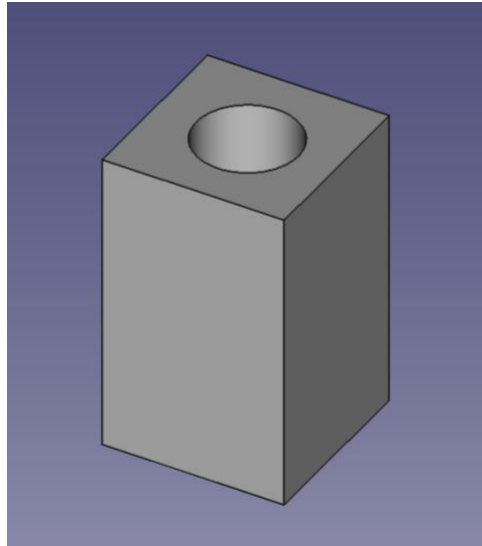

Figure 16 Cubes Designed to Hold the LED matrix PCBs

The ships were also designed in FreeCAD and were designed to the size of the array of the LEDs and sensors. Due to the vertical distance between each group of LEDs and sensors being different than the horizontal distance between each group of LEDs and sensors, two separate groups of ships were created. Both groups of ships include one 4-unit length, two 3-unit length, and two 2-unit length ships but they are sized differently. If the player wanted to place a ship horizontally on the array, then the player would select one of the ships from the horizontally sized ships and vice versa. The ships were also 3D printed with the Ultimaker S3 printer and are shown in figure 17.
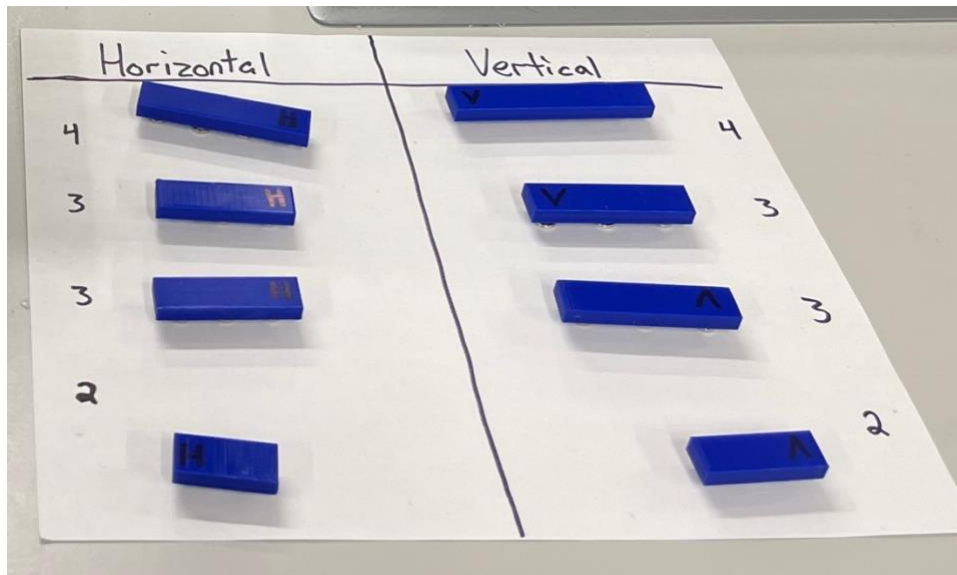

Figure 17 3D Printed Ships

Assembly

To separate the light emitting from the LEDs a grid for each array of LEDs was created. The height of the grid had to be small the only saw available at the time was a jigsaw. A jigsaw is not stable enough to cut slim pieces of wood which is why it was decided to make the grids out of popsicle sticks. The popsicle sticks were cut and glued together to create the grids. The size of each grid was different due to the size of each array being different.

Plexiglass was cut to the size of the box while excluding the thickness of the box so that the plexiglass fit inside and was flush with the top of the box. The plexiglass was cut using a plastic cutter and sanded down so that it was smooth. The purpose of the plexiglass was to have a flat surface to place the ships on while being able to see the light from the LEDs. Wooden dowels were purchased to hold the LED matrix PCBs to the cubes and to secure the plexiglass by gluing the two together.

Since the number of required ships doubled, the number of magnets purchased was not enough. Before finalizing which magnets that were going to be used other magnets were purchased. Multiple types of magnets were purchased and then a specific type of magnet was chosen to use because it worked well with the sensors at the desired height. To provide a magnetic field strong enough, the old magnets were glued together and were glued onto the ships.

To start the assembly process, four cubes were placed on the bottom surface of the device and glued in place. The PCB containing the LEDs and the sensors was placed on top of the four cubes and glued together by the corners of the PCBs. The grid was placed on top of the array of LEDs and sensors and a small amount of glue was used to ensure that it would stay in place. Four cubes were then placed on the top surface of the device and glued in place. The PCB containing just the LEDs was glued to the four cubes and the grid was secured on to the array of LEDs by a small amount of glue. A hole was drilled in the power supply box so that the wire that plugs in to a wall socket could be plugged in and connected to the power supply PCB. Two wide holes were drilled into the back of the device to allow the ribbon cables to be connected to the LED matrix PCBs from the power supply. The power supply PCB was glued inside the power supply box and then the power supply box was glued to the back of the device. The ribbon cables were connected to the LED matrix PCBs and then each plexiglass piece was placed on top of each grid. The plexiglass pieces were secured onto the edges of the device by glue. The final product of the assembly is shown in figure 18.
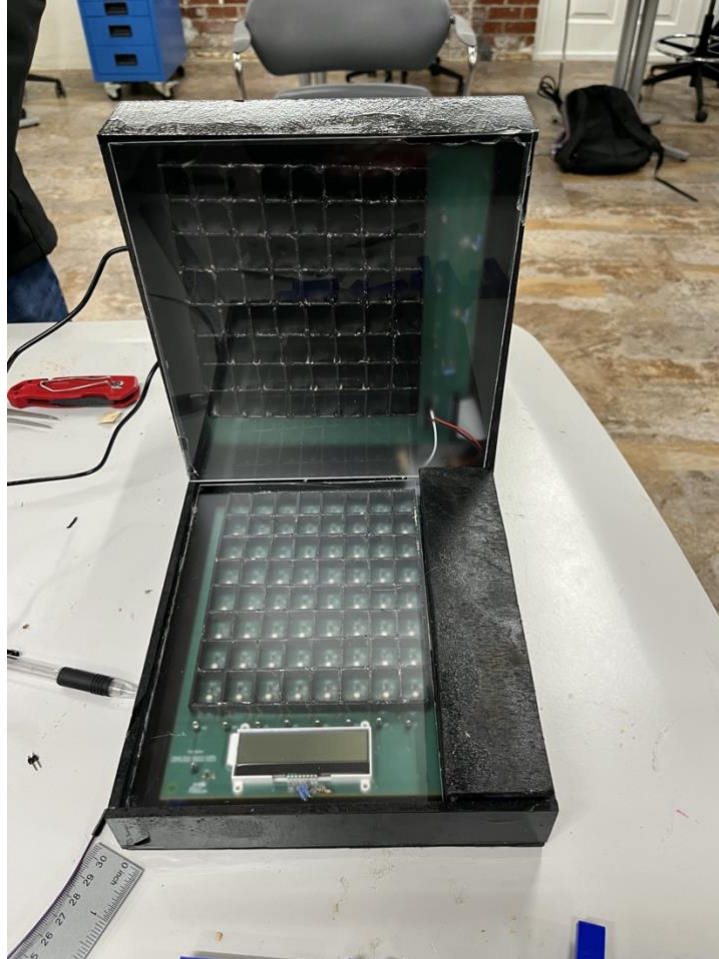
Figure 18 The Final Product

Project Time Line

The first proposed project timeline can be seen in the Gantt chart in Figure 19. The Gantt chart is categorized in the following subjects: Administrative (gray), Software (yellow), Assembly (red), PCB (purple), Sensors (blue), and System Testing (pink). Initially we had optimistically planned for each phase of the project to go along much faster than they had. In particular the software and Hardware took significantly longer to conduct. Our schematic and therefore PCB took us a while to get the correct functionality desired. Because it took us much longer to get our PCB boards, testing for the sensors was pushed back significantly from our original goal. Figure 20 shows our final Gantt chart reflective of how long certain tasks took us within the course. As mentioned prior there were many things that took us far longer to complete than originally scoped. Another aspect that slowed down our progress significantly were the unfortunate events of November 14, 2022. This rattled our team and progress slowed to a halt as we processed the events and went home to spend time with our families. Once we returned progress returned as scheduled, but we had lost a significant amount of working time. This caused a lot of our progress to be pushed towards the deadline, which deviated greatly from our original plan.
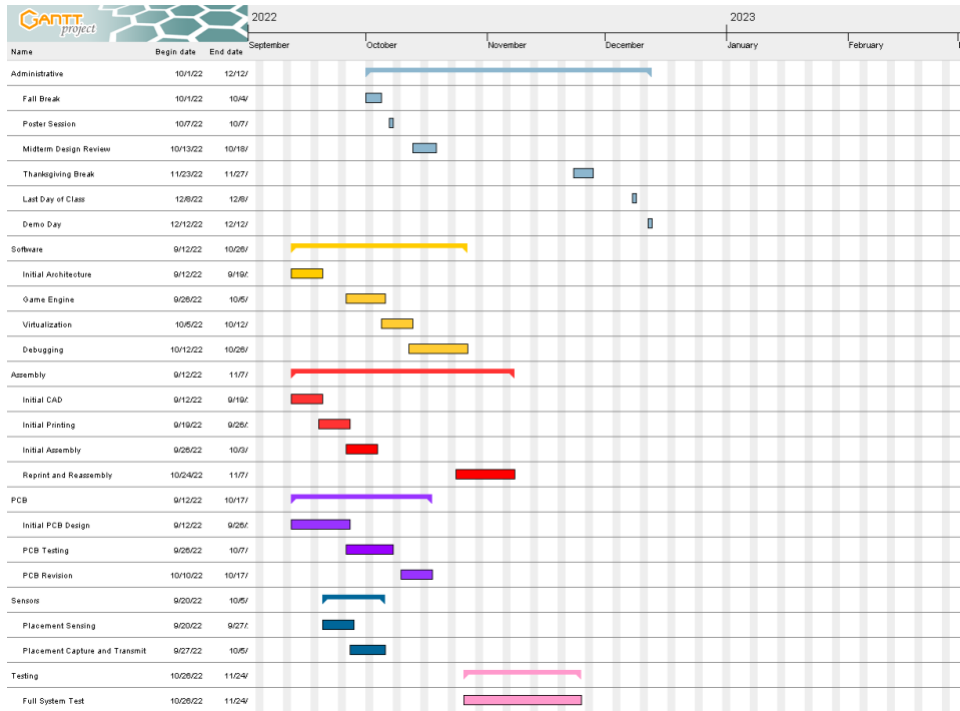
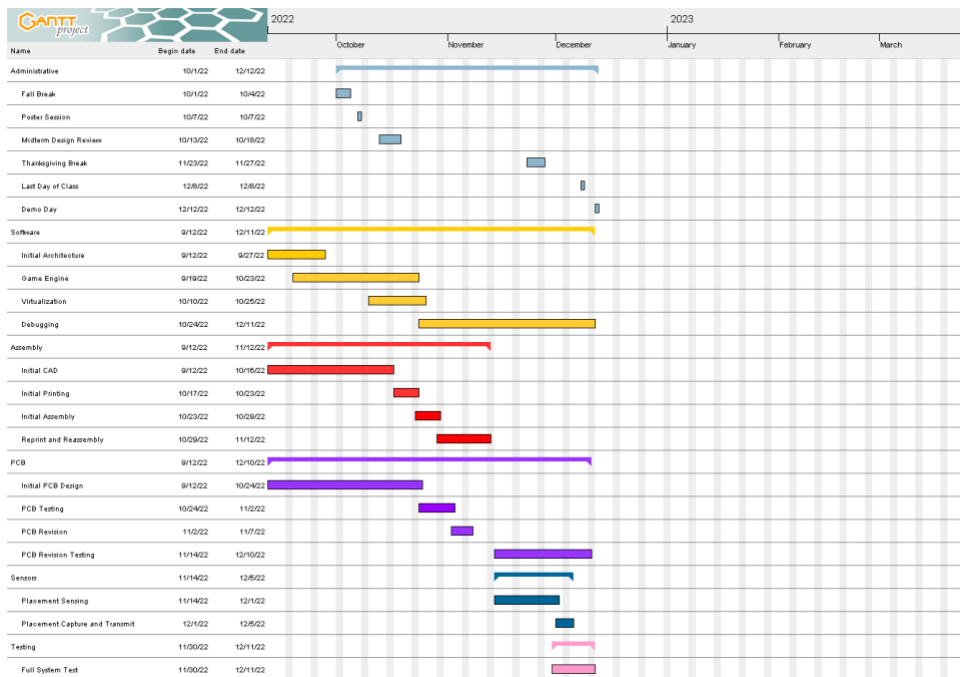**Figure 19 : Initial Proposed Gantt chart**

Figure 20 : Final Gantt chart

Maggie's primary responsibility was the design and build of the physical device and parts of the game. This included designing and building the device, power supply box, grids, cubes, ships, and plexiglass. Her secondary was assisting with the PCB design.

Deianira's primary responsibility was PCB and Schematic design. This included choosing components that met our specifications and constraints. Her secondary was assisting with the hardware assembly.

Kyle's primary responsibility was the embedded coding that communicated with the three main peripherals: LEDs, Hall-Effect Sensors, and LCD. His secondary was assisting with the software for the Battleship game logic.

Madisen's primary responsibility was software for the Battleship game logic. Her secondary was assisting with the embedded coding.

Primary and Secondary responsibilities were designed such that each member could easily reach for assistance from a designated pair. It also allowed for both hardware and software to be worked on simultaneously without delay. This allowed us to meet up at the end and combine the two (hardware and software) even with the significant time loss this semester.

Test Plan

Hardware Testing

Our hardware testing consisted primarily of ensuring the different components worked individually. To start we ensured that our sensors could in fact sense our magnets. This was tested by placing a singular magnet over various sensors and reading their output in Code Composer Studio. Should they read a 0 we knew they were operating as intended. Next was testing our LCD, to do this simple strings were passed via SPI to the LCD. Our final hardware test was to ensure our multiplexing chips and LEDs could operate. These tests were done by communicating with the multiplexing chips our desired color, and telling them to turn on the LEDs. color, and telling them to turn on the LEDs.

Software Testing

In order to test the Battleship Code, the testing was broken into two parts. The first part was completing the logic for the game in the code. This was tested by taking inputs from a player for ship placement and target selection in the terminal. The computer and player's boards as the game progressed were displayed in the terminal as well. Invalid selection and computer smart logic was also tested by playing a full game communicating via the terminal.

Once the Battleship Code was tested the second step was testing the embedded Coding. This included testing the communication between the code and the hardware. This was done by connecting to the MSP. First the communication between the LEDs and the code were tested by turning on all LEDs at once and by testing turning on one LED in a cell. After testing LEDs, the Hall Effect Sensors were tested to detect a magnet in the code. This was done by placing a magnet in a cell and detecting that magnet in software. The last step was interfacing with the LCD. This was tested by sending strings to the LCD display. The display string was then compared to the string encoded.

Full System Testing

The Full System was tested in two main sections: player input identification, and correct flow of the game and gameboards. First, we tested that the code could correctly identify all of the ships which include reading several magnet values. This was tested by placing four magnets down at a time in different locations repeatedly to ensure that the code correctly identified all locations. This was then repeated with three and two magnets. After ensuring that the player's inputs were correctly identified, we tested displaying those locations with the LEDs. After correctly communicating with the LEDs, we verified that the Battleship game operated correctly. To test the games operations, multiple games were played. While playing the games the following things were checked: ship placement, random and smart moves by the computer player, LEDs displayed, player target selection inputs, LCD communication with player, and winner of the game.

Final Results

We successfully built a complete device that allows for a player to play battleship against a computer on a physical device. The system contains all key components including LCD display, 128 LEDs and a board with 65 Hall Effect Sensors for the player to interact with. All eight requirements set were fulfilled. The first requirement was having a LED array that properly displays the robot's hits and misses. The device's top LED display worked correctly when displaying the robot's hits and misses in different colors. The Second requirement was having a LED array that properly displays the user's hits and misses. The devices bottom LED display worked correctly when displaying the players hits and misses. The third requirement was a strategy system for the moves performed by the robot is successfully implemented bot just random selections. The code successfully implemented an AI that allowed the computer to make specific moves based off of results from previous selections. The fourth requirement was that the player is able to input their ships' location at the beginning of the game. This was executed correctly with the Hall Effect Sensors and the embedded coding to recognize where a player had placed its ships. The fifth requirement was that the player is able to successfully input their move without error. The player was able to use a game piece to select a specific cell target, if the player made an invalid move the LCD would notify the player to try again. The sixth requirement was that the game can be reset/restarted at any point during the game. This was successfully executed by the player placing a game piece on the confirm sensor on the board during the computer's turn. When this was done the game automatically reset to the beginning of the game. The seventh requirement is that the system displays updates to the user regarding instructions, the robot's current move, whether the player won or lost. The LCD screen on the device successfully displayed instructions to the player like "Place your 4 long ship." It also communicated when it was the computer's move and the result of that move. At the end of the game the screen informed the player who won the game. The eighth requirement was that the robot's ships are displayed for any spectators and not visible to the player. This requirement was complete by showing the state of the computer's game board on a laptop screen. The proposed Grades for requirements met is shown in the table.

Letter Grade	Criteria
A+	All 8 requirements are met
A	7 requirements are met

A-	6 requirements are met
B+	5 requirements are met
B	4 requirements are met
C+	3 requirements are met
C	2 requirements are met
D	1 requirement is met
F	0 requirements are met

Table 2 Grading Rubric Key

Costs

The cost to produce the battleship game this semester was greater than the cost of one production model due to incorrect orders. The university offers free 3D printing for students so our 3D printed components were free but would be a cost if this game was produced in large quantities. Table 3 shows a high-level breakdown of the total costs if the battleship game was made in limited and large quantities.

	Cost for 1 unit	Cost for 10000 unit
MSP430	$15.59	$15.59
Multiplexing Chips	$11.19	$7.995
Multiplexer Chips	$0.50	$0.16452
LCD Display	$12.73	$10.716
Voltage Regulator	$1.97	$0.93
Wall mount	$10.88	$7.84
Barrel jack	$0.87	$0.43
Test Points	$0.42	$0.17
LEDs	$0.36	$0.277
Magnetic Sensors	$0.49	$0.19
PCB	$33	$5.75
Magnets	$4.20	$3.30
Ribbon Cables	$22.82	$22.82
Total	$115.02	$76.17

Table 3 Automated Battleship costs

The most expensive part of the system was the PCB and the ribbon cables. If the game was scaled for mass production in 10,000 units, the total cost for one unit would decrease by 33.78% from $115.02 to $76.17. To decrease costs in mass production, the 3D printed components could be manufactured with automated equipment instead of using 3D printers, which would decrease the cost for the ships and the cubes.

Future Work

We believe the project could be improved upon by ensuring the grids for the sensors and LEDs were actually squares rather than a rectangle. This would cut down on the number of game pieces needed and would allow for a clearer game play experience. We would also suggest using

a coding language that supported a truer randomization function, which C currently lacks. The project could be expanded upon by allowing the computer to remember wins and losses and record that data so that the player could improve upon their strategy, and the option to choose a difficulty level.

It would be our recommendation to communicate and collaborate early on different parts of the project. This is to ensure that your work is being checked by someone else within your group, and can help you avoid potential glaring mistakes. One such instance being when our PCB design was checked over by another group member, doing so allowed us to notice a missing trace and ground pin that would have severely hindered our project's process.

We also did not foresee how difficult it would be to choose magnets for our sensors, nor how difficult it would be to integrate our software with our hardware. Both of these were things we had projected to be rather simple, but instead needed multiple revisions to get right. Should we redo this process we would start the process for both of these sooner. To allow for mistakes and breaks along the way.

References

[1]   M. Barr and A. Copeland-Stewart, "Playing video games during the COVID-19 pandemic and effects on players' well-being," *Games and Culture*, vol. 17, no. 1, May 2021.

[2]   F. Sumac, I. Aleksi and Ž. Hocenski, "Digital chess board based on array of Hall-Effect sensors," 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1011-1014, doi: 10.23919/MIPRO.2017.7973572.

[3]   "Electronic Battleship," Hasbro. [Online]. Available: https://shop.hasbro.com/en-us/product/electronic-battleship:85D65E5E-5056-9047-F5BA-D5829EC8FFEC [Accessed: 23-Sep-2022]

[4]   "CCStudio", *CCSTUDIO IDE, configuration, compiler or debugger | TI.com* [Online]. Available: https://www.ti.com/tool/CCSTUDIO. [Accessed: 12-Sep-2022]

[5]   "FreeCAD", *FREECAD*[Online]. Available: https://www.freecadweb.org/index.php. [Accessed: 23-Sep-2022]

[6]   "Kicad", *Kicad | Digikey.com* [Online]. Available: https://www.kicad.org/. [Accessed: 22-Sep-2022]

[7]   "2 & 4 Layer PCB Special Pricing Options," Advanced Circuits, [Online]. Available: https://www.4pcb.com/pcb-prototype-2-4-layer-boards-specials.html. [Accessed 12 December 2022].

[8]   J. Nayak, M. Mishra, B. Naik, H. Swapnarekha, K. Cengiz, and V. Shanmuganathan, "An impact study of COVID-19 on six different industries: Automobile, energy and Power,

agriculture, education, travel and Tourism and Consumer Electronics," Expert systems, Mar-2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8014102/. [Accessed: 12-Sep-2022].

[9]   "DigiKey electronics Home," DigiKey Electronics - Electronic Components Distributor. [Online]. Available: https://www.digikey.com/. [Accessed: 10-Sep-2022].

[10]   "Mouser electronics Home," Mouser Electronics - Electronic Components Distributor. [Online]. Available: https://www.mouser.com/. [Accessed: 22-Sep-2022].

[11]   "FreeDFM" FreeDFM- A service of advanced circuits [Online]. Available: https://www.my4pcb.com/net35/FreeDFMNet/FreeDFMHome.aspx. [Accessed: 11-Oct-2022].

[12] "WWW Electronics Incorporated." WWW Electronics Inc – PCB Manufacturer. [Online]. Available: https://3welec.com/ [Accessed Dec. 14, 2022].

[13]   "Ultimaker S3," https://ultimaker.com. [Online]. Available: https://ultimaker.com/3d-printers/ultimaker-s3?utm_campaign=2022_Alwayson_srengineer_traffic_do_US&utm_source=google&utm_medium=cpc&gclid=Cj0KCQiA4uCcBhDdARIsAH5jyUlY8qf0vTBLIHYF0KfkFC2Y4zPwx9QFKQdrLg2lmXMDOgu-BEsrNK0aAh-QEALw_wcB. [Accessed: 11-Dec-2022].

[14]   "MSP430F5529 | MSP430x5xx and MSP430x6xx | MSP430 ultra-low-power MCUs | Description & parametrics." [Online]. Available: https://www.ti.com/product/MSP430F5529. [Accessed: 01-Oct-2022].

[15]   "About the U.S. Electricity System and its Impact on the Environment," *EPA*, 11-Jul-2022. [Online]. Available: https://www.epa.gov/energy/about-us-electricity-system-and-its-impact-environment. [Accessed: 25-Sep-2022].

[16]   S. Hall, "Is 3D printing increasing the plastic pollution problem?," Azocleantech.com, 03-Apr-2019. [Online]. Available: https://www.azocleantech.com/article.aspx?ArticleID=870. [Accessed: 10-Dec-2022].

[17]   C. Smith, "Huntron news and information," Huntron News and Information, 02-Jul-2020. [Online]. Available: https://huntron.com/news/pcb-repair-more-environmentally-friendly/. [Accessed: 10-Dec-2022].

[18]   "67730 Federal Register /vol. 73, no. 222/Monday, November 17, 2008 ...," *Federal Reserve*, 17-Nov-2008. [Online]. Available: https://www.cpsc.gov/s3fs-public/pdfs/blk_pdf_lrtgafin.pdf. [Accessed: 26-Sep-2022].

[19]   "Part 1505 - Requirements for Electrically Operated Toys or Other Electrically Operated Articles Intended for Use by Children," Federal Register, 08-Sep-2022. [Online]. Available:

https://www.ecfr.gov/current/title-16/chapter-II/subchapter-C/part-1505. [Accessed: 12-Sep-2022].

[20]   Sagepub.com. [Online]. Available:
https://journals.sagepub.com/doi/full/10.1177/15554120211017036 [Accessed: 10-Dec-2022].

[21]   Apa.org. [Online]. Available: https://www.apa.org/news/press/releases/2022/05/covid-19-increase-loneliness#:~:text=The%20researchers%20found%20a%20small,all%20groups%20experienced%20that%20increase. [Accessed: 10-Dec-2022].

[22]   M. Barr, *Embedded C coding standard*. Estats Units d'Amèrica: Barr Group, 2018.

[23]   "NEMA enclosure types," Wikipedia, 19-Jun-2022. [Online]. Available:
https://en.wikipedia.org/wiki/NEMA_enclosure_types#:~:text=The%20National%20Electrical%20Manufacturers%20Association,type%2Ddependent%20designated%20environmental%20conditions. [Accessed: 12-Sep-2022].

[24]   "IPC Standards," *IPC International, Inc.*, 03-Jun-2022. [Online]. Available:
https://www.ipc.org/ipc-standards. [Accessed: 12-Sep-2022].

[25] L. Coffin, "US1988301A - game board," *Google Patents*, 15-Jan-1935. [Online]. Available:
https://patents.google.com/patent/US1988301A/en. [Accessed: 12-Dec-2022].

[26] M. Maharbiz, S. Jaqua, and T. Morichau - Beauchant, "US10456660B2 - board game with dynamic characteristic tracking," *Google Patents*, 29-Oct-2019. [Online]. Available:
https://patents.google.com/patent/US10456660B2/en?q=sensor&before=priority%3A20080603&scholar. [Accessed: 12-Dec-2022].

[27] Patent 3

[28]   "L7805CV" *Digikey.com.* [Online]. Available:
https://www.digikey.com/en/products/detail/stmicroelectronics/L7805CV/585964. [Accessed: 07-Oct-2022].

[29]   "LD1117AV33" *Digikey.com.* [Online]. Available:
https://www.digikey.com/en/products/detail/stmicroelectronics/LD1117AV33/586006?s=N4IgTCBcDaIDIBECMKDsBBAagZmyAugL5A. [Accessed: 07-Oct-2022].

[30]   "AH3582-SA-7" *Digikey.com.* [Online]. Available:
https://www.digikey.com/en/products/detail/diodes-incorporated/ah3582-sa-7/9649761. [Accessed: 07-Oct-2022].

[31]   "SN74HC151DR" *Digikey.com.* [Online]. Available: https://www.digikey.com/en/products/detail/texas-instruments/SN74HC151DR/562899. [Accessed: 07-Oct-2022].

[32]   "NHD-C0220BIZ" *Digikey.com.* [Online]. Available: https://www.digikey.com/en/products/detail/newhaven-display-intl/NHD-C0220BIZ-FSW-FBW-3V3M/2626407. [Accessed: 07-Oct-2022].

[33]   "MAX7219" *Digikey.com.* [Online]. Available: https://www.digikey.com/en/products/detail/analog-devices-inc-maxim-integrated/max7219cwg-t/1514450. [Accessed: 07-Oct-2022].

[34]   "CLX6F-FKC" *Digikey.com.* [Online]. Available: https://www.digikey.com/en/products/detail/creeled-inc/CLX6F-FKC-CKNNQDGBB7A363/6355679?s=N4IgTCBcDaIMIBkAaA2AYgWjQaThu2AcoQIoAiA4gEJUDsAggMwqMgC6AvkA. [Accessed: 07-Oct-2022].

Appendix

Appendix A

	Cost per Unit	Total Cost
PCB	$33	$99
LCD Display	$12.73	$12.73
5V Voltage Regulator	$1.13	$1.13
3.3V Voltage Regulator	$0.84	$0.84
Wall Mount	$10.88	$10.88
Barrel Jack	$0.87	$0.87
Test Points	$0.34	$17.04
Hall Effect Sensors	$0.2538	$25.38
Multiplexer Chips	$0.425	$4.25
LEDs	$0.258	$38.7
Multiplexing Chips	$10.287	$102.87
MSP430	$15.59	$15.59
Ribbon Cables	$22.89	$45.78
Plexiglass	$8.18	$16.36
Popsicle Sticks	$5.00	$5.00
Magnets		$20.00
Total		$416.42

Table 4 Detailed Spreadsheet of Costs